CPA/2785

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

ASSISTANT COMMISSIONER FOR PATENTS
BOX CPA
Washington, D.C. 20231

Attorney's Docket Number: **06502.0018-01**

Prior Application: Art Unit: **2785**

Examiner: **Dieu-Minh Thai LE**

RECEIVED
NOV 24 2000
Technology Center 21

OIPE JC68 NOV 20 2000 PATENT & TRADEMARK OFFICE

SIR: This is a request for filing a

X Continuation or ☐ Divisional application under 37 C.F.R. § 1.53(b) of pending prior nonprovisional application Serial No. 09/023,074 filed February 13, 1998 of:

Inventor	Title of Invention
1. Vladimir Matena	METHOD AND APPARATUS FOR RELIABLE DISK FENCING IN A MULTICOMPUTER SYSTEM

Match & Return

1. **X** Enclosed is a complete copy of the prior application including the oath or Declaration and drawings, if any, as originally filed. I hereby verify that the attached papers are a true copy of prior application Serial No. 09/023,074 as originally filed on February 13, 1998.

2. ☐ Enclosed is a substitute specification under 37 C.F.R. §1.125.

3. ☐ Cancel Claims________________.

4. **X** A Preliminary Amendment is enclosed.

5. **X** The filing fee is calculated on the basis of the claims existing in the prior application as amended at 3 and 4 above.

11/21/2000 EHAMMOND 00000030 09023074
01 FC:131 710.00 CP

Adjustment date: 05/08/2001 CFLORES
11/21/2000 EHAMMOND 00000030 09023074
01 FC:131 -710.00 OP

LAW OFFICES
NEGAN, HENDERSON,
ARABOW, GARRETT
& DUNNER
300 I STREET, N.W.
SHINGTON, DC 20005
202-408-4000

Basic Application Filing Fee					$710	$ 710.00
	Number of Claims		Basic	Extra Claims		
Total Claims		-	20		x $18	
Independent Claims		-	3		x $78	
[ ] Presentation of Multiple Dep. Claim(s)					+$260	
					Subtotal	$ 710.00
					Reduction by 1/2 if small entity	-
					TOTAL APPLICATION FILING FEE	$ 710.00

6. X A check in the amount of $ 710.00 to cover the filing fee is enclosed.

7. X The Commissioner is hereby authorized to charge any fees which may be required including fees due under 37 C.F.R. § 1.16 and any other fees due under 37 C.F.R. § 1.17, or credit any overpayment during the pendency of this application to Deposit Account No. 06-0916.

8. X Amend the specification by inserting before the first line, the sentence:

--This is a continuation of application Serial No. 09/023,074, filed February 13, 1998, which is a continuation of application Serial No. 08/552,316, filed November 2, 1995, incorporated herein by reference.--

9. ☐ New formal drawings are enclosed.

10. X The prior application is assigned of record to: Sun Microsystems, Inc.

11. ☐ Priority of application Serial No. ________ filed on ____________ ____ in ________________ (country) is claimed under 35 U.S.C. § 119. A certified copy

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

Attorney Docket No. **06502.0018-01**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Continuation Application of:

MATENA, Vladimir

Serial No.: 09/023,074

Filed: February 13, 1998

For: METHOD AND APPARATUS FOR RELIABLE DISK FENCING IN A MULTICOMPUTER SYSTEM

Prior Application Data
Serial No. 08/552,316
Filed: November 2, 1995

Group Art Unit: 2785

Examiner: LE, Dieu-Minh Thai

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

PRELIMINARY AMENDMENT

Prior to the examination of the above-reference application, please amend this application as follows:

IN THE CLAIMS:

Please amend claims 1, 14, 17, and 24 as follows:

1. (Amended) A method for preventing access to a shared peripheral device by a processor-based node in a multinode system, [including the steps of] comprising:

(1) storing at the peripheral device a first unique value representing a first configuration of the multinode system;

(2) sending an access request from the node to the device, the request including a second unique value representing a second configuration of the multi[-]node system;

LAW OFFICES
NNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

(3) determining whether said first and second values are identical; [and]

(4) if the first and second values are identical, then executing the access request to the peripheral device; and

repeating steps (3) and (4) each time an access request is sent from the node to the device.

14. (Twice Amended) A computer usable medium having computer readable code embodied therein for preventing access to a shared peripheral device by a processor-based node in a multinode system, the computer usable medium comprising:

a storage module configured to store a first unique value representing a first configuration of the multinode system;

a reception module configured to receive [an] access [request] requests from a node to the shared peripheral device, [the] each access request including a second unique value representing a second configuration of the multinode system;

a comparator module configured to determine, for each access request received, whether said first and second values are identical; and

an execution module for executing [the] each access request at the peripheral device, if the first and second values are identical.

17. (Twice Amended) A computer usable medium having computer readable code embodied therein for preventing access to a shared peripheral device by a processor-

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

based node in a multinode system having a plurality of nodes, the [resource] shared peripheral device being coupled to the system by a resource controller, the computer usable medium comprising:

a membership monitor module configured to determine a membership list of the nodes including said [resource] shared peripheral device, on the system at predetermined times, including at least at a time when the membership of the system changes;

a resource manager module configured to determine when the [resource] shared peripheral device is in a failed state and for communicating the failure of the [resource] shared peripheral device to said membership monitor to indicate to the membership monitor to generate a new membership list;

a configuration value module configured to generate a unique value based upon said new membership list and to store said unique value locally at each node on the system; and

an access control module configured to block access requests by at least one requesting node to said [resource] shared peripheral device when the locally stored unique value at said requesting node does not equal the unique value stored at said resource controller.

24. (Twice Amended) A computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by a remote computer,

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

causes the remote computer to perform the steps of:

storing at a peripheral device a first unique value representing a first configuration of a multinode system;

sending an access request from a node to the shared peripheral device, the request including a second unique value representing a second configuration of the multinode system;

determining whether said first and second values are identical; [and]

executing the access request at the peripheral device if the first and second values are identical; and

repeating the determining step and the executing step each time an access request is sent from the node to the device.

REMARKS

Applicant has amended claims 1,14, 17 and 24 to better reflect differences between the claimed invention and the prior art. Claim 17 has been amended to correct inadvertent errors of a typographical, grammatical or clerical nature.

In the Final Office Action in the parent application, Serial No. 09/023,074, the Examiner rejected claims 1, 14-17, and 19-26 under 35 U.S.C. § 103(a) as obvious over Frey et al., U.S. Patent No. 5,416,921, in view of Kuriyama, U.S. Patent No. 5,202,923. Applicant traverses these rejections and requests the reconsideration and allowance of pending claims 1, 14-17, and 19-26.

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

The claimed determining and executing steps are not taught in the references.

Regarding the rejections of independent claims 1, 14, and 24 as obvious over Frey et al. in view of Kuriyama, the references do not show each claimed feature. Applicant has amended independent claims 1, 14, and 24 to more distinctly point out and claim the present invention. As these amendments make clear, for each access request received, it is determined whether the first and second values are identical and access to the shared peripheral device is executed or not executed based on the determination. The determination, using the first and second values, is made every time an access request is received from a node.

In contrast, as the Examiner noted, Frey et al. does not teach first and second values. Although Kuriyama discloses a first check and a second check, these are performed selectively, not every time a program registration method is invoked. In Kuriyama, if the first check determines that the program is not registered, registration can occur and the second check (i.e., validation of the data) is not performed. This is distinct from first and second unique values of claims 1, 14, and 24, which are used each time a node attempts to access a peripheral device. The first and second unique values are always used together to determine access to the peripheral device.

Because the references do not teach or suggest every element of claims 1, 14, and 24, as amended, Applicant respectfully requests the reconsideration and allowance of claims 1, 14, and 24.

LAW OFFICES
INNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

The first and second check characters of Kuriyama are patentably distinct from the claimed first and second unique values.

Further regarding the rejections of independent claims 1, 14, and 24 as obvious over Frey et al. in view of Kuriyama, the references do not show each claimed feature. Noting that Frey et al. does not disclose first and second unique values, the Examiner stated that Kuriyama teaches "first and second check character stored in memory used to determining the validity of the program via a comparison capability," making a modification to Frey et al. obvious. However, the first and second check characters in Kuriyama are patentably distinct from the first and second unique values in the claimed invention, and it would not have been obvious or desirable to one of ordinary skill in the art to modify or combine Frey et al. and Kuriyama as suggested.

First of all, the first and second check characters in Kuriyama are used to test two different things. The first check is for whether a program has been registered in memory. The second check is for whether the program data is valid. The program can only be registered in memory if the program is not already registered or if the data is invalid. This is distinguishable from the claimed first and second unique values. The first value is stored by a peripheral device. The second value is passed to the peripheral device from a node. If the first value and the second value are identical, the node can access the peripheral device. Therefore, the first and second unique values in claim 1 are used together to test the same thing (access to a peripheral device) rather than separately to test two different things (program registration and data validity)

LAW OFFICES
INNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

as taught by Kuriyama.

Secondly, the second check in Kuriyama is only tested after the first check determines that the program is registered. If the first check determines that the program is not registered, registration can occur and there is no need for the second check. This is different than the claimed first and second unique values. In the present invention, both values are used each time a node attempts to access a peripheral device. The second unique value is always used, together with the first unique value, to determine access.

Third, in Kuriyama, the second check character is calculated by the checking means as a function of the first check character. This is distinct from claims 1, 14, and 24, where the first value is stored at the peripheral device and the second value is passed to the device from a node. The first value is not used to determine the second value, and the peripheral device does not generate the second value.

The only similarity between the claimed first and second unique values and the first and second check characters taught in Kuriyama is that there are two values. This does not make it obvious to modify Frey et al. as suggested. Therefore, Applicant respectfully requests that the rejections of claims 1, 14, and 24 be reconsidered and withdrawn.

Claims 15 and 16 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 14 as well as their additional recitations. Claims 25 and 26 are patentably distinct from the cited prior art for at least the reason of their

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

dependence from claim 24 as well as their additional recitations. Therefore, Applicant respectfully requests the reconsideration and withdrawal of the rejections of claims 15-16 and 25-26.

The claimed resource manager is neither disclosed nor suggested by the references.

Regarding the rejection of claim 17 as obvious over Frey et al., in view of Kuriyama, the references do not show each feature of the claim. Neither reference teaches or suggests a resource manager module configured to determine when the shared peripheral device is in a failed state. Once it determines that the shared peripheral device is in a failed state, the claimed resource manager communicates the failure to the membership monitor to generate a new membership list.

As the Examiner noted, Frey et al. teaches a resource manager that, unlike in claim 17, receives a fence request against a failed member of the system. (col. 8, lines 40-46). The fence request in Frey et al. is issued by an operating system when that operating system detects a failure in one of its own subsystems. (col. 8, lines 35-40). The resource manager in Frey et al. does not detect the failure, nor does it notify a membership monitor to generate a new membership list. Therefore, the references do not teach or suggest the elements of claim 17. Applicant respectfully requests that the rejection of claim 17 be reconsidered and withdrawn.

Claims 19-23 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 17 as well as their additional recitations.

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

The claimed configuration value module is neither disclosed nor suggested by the references.

Further regarding the rejection of claim 17 as obvious over Frey et al., in view of Kuriyama, the references do not show each feature of the claim. Frey et al. does not disclose a configuration value module configured to generate a unique value based upon a new membership list and to store the unique value locally at each node on the system. The Examiner stated that Kuriyama teaches "an electronic device or computer device having capability to prevent program being illegally written after registration" and "first and second check character stored in memory used to determining the validity of the program via a comparison capability," making a modification to Frey et al. obvious.

However, the configuration value module is not taught or suggested by the references. The first check character in Kuriyama is used to determine whether a program is already registered in memory, and the second check character tests whether the program data is valid. This is patentably distinct from the configuration value module of claim 17. The claimed module generates a unique value based upon the nodes that can access a shared resource and stores the unique value locally at each of the nodes. The claimed element is different in both form and function from the check characters taught by Kuriyama. Therefore, it was not obvious to modify Frey et al. as suggested. Applicant respectfully requests that the rejection of claim 17 be reconsidered and withdrawn.

Claims 19-23 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 17 as well as their additional recitations.

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

In view of the foregoing amendments and remarks, Applicant respectfully requests the reconsideration and reexamination of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: November 20, 2000

By: ______________________
Jeffrey A. Berkowitz
Reg. No. 36,743

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
ASHINGTON, DC 20005
202-408-4000

2A41

2785

Attorney Docket No. **06502.0018-02**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Continuation Application of: MATENA, Vladimir Application No.: Not yet assigned Filed: November 20, 2000 For: METHOD AND APPARATUS FOR RELIABLE DISK FENCING IN A MULTICOMPUTER SYSTEM	)	Prior Application Data Serial No. 09/023,074 Filed: February 13, 1998 Group Art Unit: 2785 Examiner: LE, Dieu-Minh Thai

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

SECOND PRELIMINARY AMENDMENT

Please cancel all amendments and remarks specified in the Preliminary Amendment filed November 20, 2000 and prior to the examination of the above-reference application, please amend this application as follows:

IN THE CLAIMS:

Please amend claim 1 and add claims14-26 as follows:

1. (Amended) A method for preventing access to a shared peripheral device by a processor-based node in a multinode system, [including the steps of] <u>comprising</u>:

(1) storing at the peripheral device a first unique value representing a first configuration of the multinode system;

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

(2) sending an access request from the node to the device, the request including a second unique value representing a second configuration of the multi[-]node system;

(3) determining whether said first and second values are identical; [and]

(4) if the first and second values are identical, then executing the access request to the peripheral device<u>; and</u>

<u>repeating steps (3) and (4) each time an access request is sent from the node to the device</u>.

--14. A computer usable medium having computer readable code embodied therein for preventing access to a shared peripheral device by a processor-based node in a multinode system, the computer usable medium comprising:

a storage module configured to store a first unique value representing a first configuration of the multinode system;

a reception module configured to receive access requests from a node to the shared peripheral device, each access request including a second unique value representing a second configuration of the multinode system;

a comparator module configured to determine, for each access request received, whether said first and second values are identical; and

an execution module for executing each access request at the peripheral device, if the first and second values are identical.

15. The computer usable medium of claim 14, wherein said storage medium

LAW OFFICES
INNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

includes a submodule configured to generate said first value using information relating to a first time when the multinode system was in said first configuration, and

further comprising a module configured to generate said second value using information relating to a second time when the multinode system was in said second configuration.

16. The computer usable medium of claim 15, wherein the comparator module includes a submodule configured to determine whether said first and second times are identical.

17. A computer usable medium having computer readable code embodied therein for preventing access to a shared peripheral device by a processor-based node in a multinode system having a plurality of nodes, the shared peripheral device being coupled to the system by a resource controller, the computer usable medium comprising:

a membership monitor module configured to determine a membership list of the nodes including said shared peripheral device, on the system at predetermined times, including at least at a time when the membership of the system changes;

a resource manager module configured to determine when the shared peripheral device is in a failed state and for communicating the failure of the shared peripheral device to said membership monitor to indicate to the membership monitor to generate a new membership list;

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

a configuration value module configured to generate a unique value based upon said new membership list and to store said unique value locally at each node on the system; and

an access control module configured to block access requests by at least one requesting node to said shared peripheral device when the locally stored unique value at said requesting node does not equal the unique value stored at said resource controller.

18. The computer usable medium of claim 17, wherein said configuration value module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

19. The computer usable medium of claim 17, wherein said membership monitor module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

20. The computer usable medium of claim 17, wherein said resource manager module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

21. The computer usable medium of claim 17, wherein said configuration module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

22. The computer usable medium of claim 17, wherein said access control module is configured to execute independently of any action by said shared resource when said shared resource is in a filed state.

23. The computer usable medium of claim 17, wherein said configuration value module includes a submodule configured to generate the unique value based at least in part upon a time stamp indicating the time at which the corresponding membership list was generated.

24. A computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by a remote computer, causes the remote computer to perform the steps of:

storing at a peripheral device a first unique value representing a first configuration of a multinode system;

sending an access request from a node to the shared peripheral device, the request including a second unique value representing a second configuration of the multinode system;

determining whether said first and second values are identical;

executing the access request at the peripheral device if the first and second values are identical;

repeating the determining step and the executing step each time an access request is sent from the node to the device.

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

25. The computer data signal for causing the remote computer to perform the step of claim 24, wherein the step of storing includes the substep of generating said first value using information relating to a first time when the multinode system was in said first configuration, and

further including the step of generating said second value using information relating to a second time when the multinode system was in said second configuration.

26. The computer data signal for causing the remote computer to perform the step of claim 25, further including the step of:

determining whether said first and second times are identical.--

REMARKS

To place the parent application, Serial No. 09/023,074, in condition for allowance, Applicant canceled claims 1, 14-17, and 19-26. Applicant filed the present continuation application on November 20, 2000, with a Preliminary Amendment addressing the Examiner's rejections of claims 1, 14-17, and 19-26. By this Preliminary Amendment, Applicant seeks to amend claim 1 and add claims 14-26 to the present continuation application.

In the Final Office Action in the parent application, Serial No. 09/023,074, the Examiner rejected claims 1, 14-17, and 19-26 under 35 U.S.C. § 103(a) as obvious over

LAW OFFICES
INNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

Frey et al., U.S. Patent No. 5,416,921, in view of Kuriyama, U.S. Patent No. 5,202,923. Applicant requests that the Examiner consider the following remarks concerning that rejection.

The claimed determining and executing steps are not taught in the references.

Regarding the rejections in the parent application of independent claims 1, 14, and 24 as obvious over Frey et al. in view of Kuriyama, the references do not show each claimed feature. Independent claims 1, 14, and 24 clearly recite that, for each access request received, it is determined whether the first and second values are identical and access to the shared peripheral device is executed or not executed based on the determination. The determination, using the first and second values, is made every time an access request is received from a node.

In contrast, as the Examiner noted, Frey et al. does not teach first and second values. Although Kuriyama discloses a first check and a second check, these are performed selectively, not every time a program registration method is invoked. In Kuriyama, if the first check determines that the program is not registered, registration can occur and the second check (i.e., validation of the data) is not performed. This is distinct from first and second unique values of claims 1, 14, and 24, which are used each time a node attempts to access a peripheral device. The first and second unique values are always used together to determine access to the peripheral device.

Because the references do not teach or suggest every element of claims 1, 14, and 24 Applicant respectfully requests allowance of claims 1, 14, and 24.

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

The first and second check characters of Kuriyama are patentably distinct from the claimed first and second unique values.

Further regarding the rejections in the parent application of independent claims 1, 14, and 24 as obvious over Frey et al. in view of Kuriyama, the references do not show each claimed feature. Noting that Frey et al. does not disclose first and second unique values, the Examiner stated that Kuriyama teaches "first and second check character stored in memory used to determining the validity of the program via a comparison capability," making a modification to Frey et al. obvious. However, the first and second check characters in Kuriyama are patentably distinct from the first and second unique values in the claimed invention, and it would not have been obvious or desirable to one of ordinary skill in the art to modify or combine Frey et al. and Kuriyama as suggested.

First of all, the first and second check characters in Kuriyama are used to test two different things. The first check is for whether a program has been registered in memory. The second check is for whether the program data is valid. The program can only be registered in memory if the program is not already registered or if the data is invalid. This is distinguishable from the claimed first and second unique values. The first value is stored by a peripheral device. The second value is passed to the peripheral device from a node. If the first value and the second value are identical, the node can access the peripheral device. Therefore, the first and second unique values in claim 1 are used together to test the same thing (access to a peripheral device) rather than separately to test two different things (program registration and data validity) as taught by Kuriyama.

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

Secondly, the second check in Kuriyama is only tested after the first check determines that the program is registered. If the first check determines that the program is not registered, registration can occur and there is no need for the second check. This is different than the claimed first and second unique values. In the present invention, both values are used each time a node attempts to access a peripheral device. The second unique value is always used, together with the first unique value, to determine access.

Third, in Kuriyama, the second check character is calculated by the checking means as a function of the first check character. This is distinct from claims 1, 14, and 24, where the first value is stored at the peripheral device and the second value is passed to the device from a node. The first value is not used to determine the second value, and the peripheral device does not generate the second value.

The only similarity between the claimed first and second unique values and the first and second check characters taught in Kuriyama is that there are two values. This does not make it obvious to modify Frey et al. as suggested. Therefore, Applicant respectfully submits that claims 1, 14, and 24 are patentable over the prior art.

Claims 15 and 16 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 14 as well as their additional recitations. Claims 25 and 26 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 24 as well as their additional recitations. Therefore, Applicant respectfully requests the allowance of claims 15-16 and 25-26.

LAW OFFICES
INNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

The claimed resource manager is neither disclosed nor suggested by the references.

Regarding the rejection in the parent application of claim 17 as obvious over Frey et al., in view of Kuriyama, the references do not show each feature of the claim. Neither reference teaches or suggests a resource manager module configured to determine when the shared peripheral device is in a failed state. Once it determines that the shared peripheral device is in a failed state, the claimed resource manager communicates the failure to the membership monitor to generate a new membership list.

As the Examiner noted, Frey et al. teaches a resource manager that, unlike in claim 17, receives a fence request against a failed member of the system. (col. 8, lines 40-46). The fence request in Frey et al. is issued by an operating system when that operating system detects a failure in one of its own subsystems. (col. 8, lines 35-40). The resource manager in Frey et al. does not detect the failure, nor does it notify a membership monitor to generate a new membership list. Therefore, the references do not teach or suggest the elements of claim 17. Applicant respectfully requests allowance of claim 17.

Claims 19-23 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 17 as well as their additional recitations.

The claimed configuration value module is neither disclosed nor suggested by the references.

Further regarding the rejection in the parent application of claim 17 as obvious over Frey et al., in view of Kuriyama, the references do not show each feature of the

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

claim. Frey et al. does not disclose a configuration value module configured to generate a unique value based upon a new membership list and to store the unique value locally at each node on the system. The Examiner stated that Kuriyama teaches "an electronic device or computer device having capability to prevent program being illegally written after registration" and "first and second check character stored in memory used to determining the validity of the program via a comparison capability," making a modification to Frey et al. obvious.

However, the configuration value module is not taught or suggested by the references. The first check character in Kuriyama is used to determine whether a program is already registered in memory, and the second check character tests whether the program data is valid. This is patentably distinct from the configuration value module of claim 17. The claimed module generates a unique value based upon the nodes that can access a shared resource and stores the unique value locally at each of the nodes. The claimed element is different in both form and function from the check characters taught by Kuriyama. Therefore, it was not obvious to modify Frey et al. as suggested. Applicant respectfully requests allowance of claim 17.

Claims 18-23 are patentably distinct from the cited prior art for at least the reason of their dependence from claim 17 as well as their additional recitations.

In view of the foregoing amendments and remarks, Applicant respectfully requests the allowance of the pending claims. Applicant further requests that the Examiner grant an interview to facilitate the examination of the present application.

Please grant any extensions of time required to enter this amendment and charge

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
SHINGTON, DC 20005
202-408-4000

any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: February 18, 2001

By: [signature]
Jeffrey A. Berkowitz
Reg. No. 36,743

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

Attorney Docket No. 82225.P758

APPLICATION FOR UNITED STATES LETTERS PATENT FOR

METHOD AND APPARATUS FOR RELIABLE DISK FENCING IN A MULTICOMPUTER SYSTEM

INVENTOR(S):

Vladimir Matena

PREPARED BY:

MATTHEW C. RAINEY, ESQ.
SUN MICROSYSTEMS, INC.
2550 Garcia Avenue, M/S PAL1-521
Mountain View, CA 94043-1100
(415) 336-0482

Method and Apparatus for Reliable Disk Fencing in a Multicomputer System

O I P E JC68
NOV 2 0 2000
PATENT & TRADEMARK OFFICE

The present invention relates to a system for reliable disk fencing of shared disks in a multicomputer system, e.g. a cluster, wherein multiple computers (nodes) have concurrent access to the shared disks. In particular, the system is directed to a high availability system with shared access disks.

RECEIVED
NOV 2 4 2000
Technology Center 2100

Background of the Invention

In clustered computer systems, a given node may "fail", i.e. be unavailable according to some predefined criteria which are followed by the other nodes. Typically, for instance, the given node may have failed to respond to a request in less than some predetermined amount of time. Thus, a node that is executing unusually slowly may be considered to have failed, and the other nodes will respond accordingly.

When a node (or more than one node) fails, the remaining nodes must perform a system reconfiguration to remove the failed node(s) from the system, and the remaining nodes preferably then provide the services that the failed node(s) had been providing.

It is important to isolate the failed node from any shared disks as quickly as possible. Otherwise, if the failed (or slowly executing) node is not isolated by the time system reconfiguration is complete, then it could, e.g., continue to make read and write requests to the shared disks, thereby corrupting data on the shared disks.

Disk fencing protocols have been developed to address this type of problem. For instance, in the VAXcluster system, a "deadman brake" mechanism" is used. See Davis, R.J., VAXcluster Principles (Digital Press 1993), incorporated herein by reference. In the VAXcluster system, a failed node is isolated from the new configuration, and the nodes in the new configuration are required to wait a certain predetermined timeout period before they are allowed to access the disks. The deadman brake mechanism on the isolated node guarantees that the isolated node becomes "idle" by the end of the timeout period.

The deadman brake mechanism on the isolated node in the VAXcluster system involves both hardware and software. The software on the isolated node is required to periodically tell the

cluster interconnect adaptor (CI), which is coupled between the shared disks and the cluster interconnect, that the node is "sane". The software can detect in a bounded time that the node is not a part of the new configuration. If this condition is detected, the software will block any disk I/O, thus setting up a software "fence" preventing any access of the shared disks by the failed node. A disadvantage presented by the software fence is that the software must be reliable; failure of (or a bug in) the "fence" software results in failure to block access of the shared disks by the ostensibly isolated node.

If the software executes too slowly and thus does not set up the software fence in a timely fashion, the CI hardware shuts off the node from the interconnect, thereby setting up a hardware fence, i.e. a hardware obstacle disallowing the failed node from accessing the shared disks. This hardware fence is implemented through a sanity timer on the CI host adaptor. The software must periodically tell the CI hardware that the software is "sane". A failure to do so within a certain time-out period will trigger the sanity timer in CI. This is the "deadman brake" mechanism.

Other disadvantages of this node isolation system are that:

- it requires an interconnect adaptor utilizing an internal timer to implement the hardware fence.
- the solution does not work if the interconnect between the nodes and disks includes switches or any other buffering devices. A disk request from an isolated node could otherwise be delayed by such a switch or buffer, and sent to the disk after the new configuration is already accessing the disks. Such a delayed request would corrupt files or databases.
- depending on the various time-out values, the time that the members of the new configuration have to wait before they can access the disk may be too long, resulting in decreased performance of the entire system and contrary to high-availability principles.

From an architectural level perspective, a serious disadvantage of the foregoing node isolation methodology is that it does not have end-to-end properties; the fence is set up on the *node* rather than on the disk controller.

It would be advantageous to have a system that presented high availability while rapidly setting up isolation of failed disks at the *disk controller*.

Other UNIX-based clustered systems use SCSI (small computer systems interface) "disk reservation" to prevent undesired subsets of clustered nodes from accessing shared disks. See, e.g., the ANSI SCSI-2 Proposed Standard for information systems (March 9, 1990, distributed by Global Engineering Documents), which is incorporated herein by reference. Disk reservation has a number of disadvantages; for instance, the disk reservation protocol is applicable only to systems having two nodes, since only one node can reserve a disk at a time (i.e. no other nodes can access that disk at the same time). Another is that in a SCSI system, the SCSI bus reset operation removes any disk reservations, and it is possible for the software disk drivers to issue a SCSI bus reset at any time. Therefore, SCSI disk reservation is not a reliable disk fencing technique.

Another node isolation methodology involves a "poison pill"; when a node is removed from the system during reconfiguration, one of the remaining nodes sends a "poison pill", i.e. a request to shut down,. to the failed node. If the failed node is in an active state (e.g. executing slowly), it takes the pill and becomes idle within some predetermined time.

The poison pill is processed either by the host adaptor card of the failed node, or by an interrupt handler on the failed node. If it is processed by the host adaptor card, the disadvantage is presented that the system requires a specially designed host adaptor card to implement the methodology. If it is processed by an interrupt handler on the failed node, there is the disadvantage that the node isolation is not reliable; for instance, as with the VAXcluster discussed above, the software at the node may itself by unreliable, time-out delays are presented, and again the isolation is at the node rather than at the shared disks.

A system is therefore needed that prevents shared disk access at the disk sites, using a mechanism that both rapidly and reliably blocks an isolated node from accessing the shared disks, and does not rely upon the isolated node itself to support the disk access prevention.

Summary of the Invention

The present invention utilizes a method and apparatus for quickly and reliably isolating failed resources, including I/O devices such as shared disks, and is applicable to a virtually any shared resource on a computer system or network. The system of the invention maintains a membership list of all the active shared resources, and with each new configuration, such as when a resource is added or fails (and thus should be functionally removed), the system generates a new epoch number or other value that uniquely identifies that configuration at that time. Thus, identi-

cal memberships occurring at different times will have different epoch numbers, particularly if a different membership set has occurred in between.

Each time a new epoch number is generated, a control key value is derived from it and is sent to the nodes in the system, each of which stores the control key locally as its own node key. The controllers for the resources (such as disk controllers) also store the control key locally. Thereafter, whenever a shared resource access request is sent to a resource controller, the node key is sent with it. The controller then checks whether the node key matches the controller's stored version of the control key, and allows the resource access request only if the two keys match.

When a resource fails, e.g. does not respond to a request within some predetermined period of time (indicating a possible hardware or software defect), the membership of the system is determined a new, eliminating the failed resource. A new epoch number is generated, and therefrom a new control key is generated and is transmitted to the all the resource controllers and nodes on the system. If an access request arrives at a resource controller after the new control key is generated, the access request will bear a node key that is different from the current control key, and thus the request will not be executed. This, coupled with preventing nodes from issuing access requests to resources that are not in the current membership set, ensures that failed resources are quickly eliminated from access, by requiring that all node requests, in order to be processed, have current control key (and hence membership) information.

The nodes each store program modules to carry out the functions of the invention -- e.g., a disk (or resource) manager module, a distributed lock manager module, and a membership module. The distribution of these modules allows any node to identify a resource as failed and to communicate that to the other nodes, and to generate new membership lists, epoch numbers and control keys.

The foregoing system therefore does not rely upon the functioning of a failed resource's hardware or software, and provides fast end-to-end (i.e. at the resource) resource fencing.

Brief Description of the Drawings

Figure 1 is a top-level block diagram showing several nodes provided with access to a set of shared discs.

Figure 2 is a more detailed block diagram of a system similar to that of Figure 1, but show-

ing elements of the system of the invention that interact to achieve disk fencing.

Figure 3 is a diagram illustrating elements of the structure of each node of Figure 2 or Figure 3 before and after reconfiguration upon the unavailability of node D.

Figure 4 is a block diagram of a system of the invention wherein the nodes access more than one set of shared disks.

Figure 5 is a flow chart illustrating the method of the invention.

Description of the Preferred Embodiments

The system of the invention is applicable generally to clustered systems, such as system 10 shown in Figure 1, including multiple nodes 20-40 (Nodes 1-3 in this example) and one or more sets of shared disks 50. Each of nodes 20-40 may be a conventional processor-based system having one or more processors and including memory, mass storage, and user I/O devices (such as monitors, keyboards, mouse, etc.), and other conventional computer system elements (not all shown in Figure 1), and configured for operation in a clustered environment.

Disks 50 will be accessed and controlled via a disk controller 60, which may include conventional disk controller hardware and software, and includes a processor and memory (not separately shown) for carrying out disk control functions, in addition to the features described below.

The system of the invention may in general be implemented by software modules stored in the memories of the nodes 20-40 and of the disk controller. The software modules may be constructed by conventional software engineering, given the following teaching of suitable elements for implementing the disk fencing system of the invention. Thus, in general in the course of the following description, each described function may be implemented by a separate program module stored at a node and/or at a resource (e.g. disk) controller as appropriate, or several such functions may be implemented effectively by a single multipurpose module.

Figure 2 illustrates in greater detail a clustered system 70 implementing the invention. The system 70 includes four nodes 80-110 (Nodes A-D) and at least one shared disk system 120. The nodes 80-110 may be any conventional cluster nodes (such as workstations, personal computers or other processor-based systems like nodes 20-40 or any other appropriate cluster nodes), and the disk system may be any appropriate shared disk assembly, including a disk system 50 as discussed in connection with Figure 1.

Each node 80-110 includes at least the following software modules: disk manager (DM),

an optional distributed lock manager (DLM), and membership monitor (MM). These modules may be for the most part conventional as in the art of clustered computing, with modifications as desired to implement the features of the present invention. The four MM modules MMA-MMD are connected in communication with one another as illustrated in Figure 2, and each of the disk manager modules DMA-DMD is coupled to the disk controller (not separately shown) of the disk system 120.

Nodes in a conventional clustered system participate in a "membership protocol", such as that described in the VAXcluster Principles cited above. The membership protocol is used to establish an agreement on the set of nodes that form a new configuration when a given node is dropped due to a perceived failure. Use of the membership protocol results in an output including (a) a subset of nodes that are considered to be the current members of the system, and (b) an "epoch number" (EN) reflecting the current status of the system. Alternatives to the EN include any time or status value uniquely reflecting the status of the system for a given time. Such a membership protocol may be used in the present system.

According to membership protocol, whenever the membership set changes a new unique epoch number is generated and is associated with the new membership set. For example, if a system begins with a membership of four nodes A-D (as in Figure 2), and an epoch number 100 has been assigned to the current configuration, this may be represented as <A, B, C, D; #100> or <MEM=A, B, C, D; EN=100>, where MEM stands for "membership". This is the configuration represented in Figure 3(a), where all four nodes are active, participating nodes in the cluster.

If node D crashes or is detected as malfunctioning, the new membership becomes <MEM=A, B, C; EN=101>; that is, node D is eliminated from the membership list and the epoch number is incremented to 101, indicating that the epoch wherein D was most recently a member is over. While all the nodes that participate in the new membership store the new membership list and new epoch number, failed node D (and another other failed node) maintains the old membership list and the old epoch number. This is as illustrated in Figure 3(b), wherein the memories of nodes A-C all store <MEM=A, B, C; EN=101>, while failed and isolated node D stores <MEM=A, B, C, D; EN=100>.

The present invention takes utilizes this fact -- i.e. that the current information is stored by active nodes while outdated information is stored by the isolated node(s) -- to achieve disk fencing. This is done by utilizing the value of a "control key" (CK) variable stored by the nodes and

the shared disk system's controller (e.g. in volatile memory of the disk controller).

Figure 4 is a block diagram of a four-node clustered system 400 including nodes 410-440 and two shared disk systems 450-460 including disks 452-456 (system 450) and 462-466 (system 460). Disk systems 450 and 460 are controlled, respectively, by disk controllers 470 and 480 coupled between the respective disk controllers and a cluster interconnect 490.

The nodes 410-440 may be processor-based systems as described above, and the disk controllers are also as described above, and thus the nodes, shared disk systems (with controllers) and cluster interconnect may be conventional in the art, with the addition of the features described herein.

Each node stores both a "node key" (NK) variable and the membership information. The NK value is calculated from the current membership by one of several alternative functions, described below as Methods 1-3. Figure 4 shows the generalized situation, taking into account the possibility that any of the nodes may have a different CK number than the rest, if that node has failed and been excluded from the membership set.

As a rule, however, when all nodes are active, their respective stored values of NK and the value of CK stored at the disk controllers will all be equal.

Node/Disk Controller Operations Using Node Key and Control Key Values

Each read and write request by a node for accessing a disk controller includes the NK value; that is, whenever a node requests read or write access to a shared disk, the NK value is passed as part of the request. This inclusion of the NK value in read and write requests thus constitutes part of the protocol between the nodes and the controller(s).

The protocol between the nodes and disk controller also includes two operations to manipulate the CK value on the controller: GetKey to read the current CK value, and SetKey to set the value of CK to a new value. GetKey does not need to provide an NK value, a CK value, or an EN value, while the SetKey protocol uses the NK value as an input and additionally provides a new CK value "new.CK" to be adopted by the controller.

The four foregoing requests and their input/output arguments may be represented and summarized as follows:

Read(NK, ...)

Write(NK, ...)

GetKey(...)

SetKey(NK, new.CK)

The GetKey(...) operation returns the current value of CK. This operation is never rejected by the controller.

The SetKey(NK, new.CK) operation first checks if the NK field in the request matches the current CK value in the controller. In the case of a match, the CK value in the controller is set equal to the value in the "new.CK" field (in the SetKey request). If NK from the requesting node doesn't match the current CK value stored at the controller, the operation is rejected and the requesting node is sent an error indication.

The Read(NK, ...) and Write(NK, ...) operations are allowed to access the disk only if the NK field in the packet matches the current value of CK. Otherwise, the operation is rejected by the controller and the requesting node is sent an error indication.

When a controller is started, the CK value is preferably initialized to 0.

Procedure Upon Failure of a Node

When the membership changes because one or more failed nodes are being removed from the system, the remaining nodes calculate a new value of CK from the new membership information (in a manner to be described below). One of the nodes communicates the new CK value to the disk controller using the SetKey(NK, new.CK) operation. After the new CK value is set, all member (active) nodes of the new configuration set their NK value to this new CK value.

If a node is not a part of the new configuration (e.g. a failed node), it is not allowed to change its NK. If such a node attempts to read or write to a disk, the controller finds a mismatch between the new CK value and the old NK value.

When a node is started, its NK is initialized to a 0 value.

Procedures for Calculating Values of the Control Key (CK)

The control key CK may be set in a number of different ways. The selected calculation will be reflected in a software or firmware module stored and/or mounted at least at the controller. In general, the calculation of the CK value should take into account the membership information:

$$CK = \text{func}(MEM, EN)$$

where: MEM includes information about the active membership list;

and EN is the epoch number.

Method 1. Ideally, the CK value would explicitly include both a list of the new membership set (an encoded set of nodes) and the epoch number. This may not be desired if the number of nodes is high, however, because the value of CK would have to include at least a bit of information for each node. That is, in a four-node configuration at least a four-bit sequence BBBB (where B = 0 or 1) would need to be used, each bit B indicating whether a given associated node is active or inactive (failed). In addition, several bits are necessary for the epoch number EN, so the total length of the variable CK may be quite long.

Method 2 and 3 below are designed to compress the membership information when calculating the CK value.

Method 2 uses only the epoch number EN and ignores the membership list MEM. For example, the CK value is set to equal the epoch number EN.

Method 2 is most practical if the membership protocol prevents network partitioning (e.g. by majority quorum voting). If membership partitioning is allowed, e.g. in the case of a hardware failure, the use of the CK value without reflecting the actual membership of the cluster could lead to conflicts between the nodes on either side of the partition.

Method 3 solves the challenge of Method 2 with respect to partitions. In this method, the CK value is encoded with an identification of the highest node in the new configuration. For example, the CK value may be a concatenation of a node identifier (a number assigned to the highest node) and the epoch number. This method provides safe disk fencing even if the membership monitor itself does not prevent network partitioning, since the number of the highest node in a given partition will be different from that of another partition; hence, there cannot be a conflict between requests from nodes in different partitions, even if the EN's for the different subclusters happen to be the same.

Of the foregoing, with a small number of nodes Method 1 is preferred, since it contains the most explicit information on the state of the clustered system. However, with numerous nodes Method 3 becomes preferable. If the system prevents network partitioning, then Method 2 is suitable.

The Method of the Invention

Given the foregoing structures and functions, and appropriate modules to implement them, the disk fencing system of the invention is achieved by following the method 510 illustrated in the flow chart of Figure 5. At box (step) 520, the membership of the clustered system is determined in a conventional manner, and the value of the membership set (or list) is stored as the value of MEM. An epoch number EN (or other unique state identifier) is generated at box 530. These two functions are carried out by the membership monitor (MM) module, which is implemented among the member nodes to determine which nodes are present in the system and then to assign a value of EN to that configuration. An example of a system that uses an MM module in this way is applicant Sun Microsystems, Inc.'s SparcCluster PDB (parallel database).

In current systems, the epoch numbers are used so that a node can determine whether a given message or data packet is stale; if the epoch number is out of date then the message is known to be have been created during an older, different configuration of the cluster. (See, for instance, T. Mann et al., "An Algorithm for Data Replication", DEC SRC Research Report, June 1989, incorporated herein by reference, wherein epoch numbers are described as being used in stamping file replicas in a distributed system.)

The present system uses the epoch number in an entirely new way, which is unrelated to prior systems' usage of the epoch number. For an example of a preferred manner of using a cluster membership monitor in Sun Microsystems, Inc.'s systems, see Appendix A attached hereto, in which the reconfiguration sequence numbers are analogous to epoch numbers. Thus, the distinct advantage is presented that the current invention solves a long-standing problem, that of quickly and reliably eliminating failed nodes from a cluster membership and preventing them from continuing to access shared disks, without requiring new procedures to generate new outputs to control the process; rather, the types of information that is already generated may be used in conjunction with modules according to the invention to accomplish the desired functions, resulting in a reliable high-availability system.

Proceeding to box 540, the node key NK (for active nodes) and control key CK are generated by one of the Methods 1-3 described above or by another suitable method.

At box 550, it is determined whether a node has become unavailable. This step is carried out virtually continuously (or at least with relatively high frequency, e.g. higher than the frequency of I/O requests); for instance, at almost any time a given node may determine that another

node has exceeded the allowable time to respond to a request, and decide that the latter node has failed and should be removed from the cluster's membership set. Thus, the step in box 550 may take place almost anywhere during the execution of the method.

Box 560 represents an event where one of the nodes connected to the cluster generates an I/O request (such as a disk access request). If so, then at box 570 the current value of NK from the requesting node is sent with the I/O access request, and at box 580 it is determined whether this matches the value of CK stored by the controller. If not, the method proceeds to step 600, where the request is rejected (which may mean merely dropped by the controller with no action), and proceeds then back to box 520.

If the node's NK value matches the controller's CK value, then the request is carried out at box 590.

If a node has failed, then the method proceeds from box 550 back to box 520, where the failed node is eliminated in a conventional fashion from the membership set, and thus the value of MEM changes to reflect this. At this time, a new epoch number EN is generated (at box 530) and stored, to reflect the newly revised membership list. In addition, at box 540 a new control key value CK is generated, the active nodes' NK values take on the value of the new CK value, and the method proceeds again to boxes 550-560 for further disk accesses.

It will be seen from the foregoing that the failure of a given node in a clustered system results both in the removal of that node from the cluster membership and, importantly, the reliable prevention of any further disk accesses to shared disks by the failed node. The invalidating of the failed node from shared disk accesses does not rely upon either hardware or software of the failed node to operate properly, but rather is entirely independent of the failed node.

Since the CK values are stored at the disk controllers and are used by an access control module to prevent failed nodes from gaining shared disk access, the disk fencing system of the invention is as reliable as the disk management software itself. Thus, the clustered system can rapidly and reliably eliminate the failed node with minimal risk of compromising the integrity of data stored on its shared disks.

The described invention has the important advantage over prior systems that its end-to-end properties make it independent of disk interconnect network or bus configuration; thus, the node configuration alone is taken into account in determining the epoch number or other unique status value, i.e. independent of any low-level mechanisms (such as transport mechanisms).

Note that the system of the invention may be applied to other peripheral devices accessed by multiple nodes in a multiprocessor system. For instance, other I/O or memory devices may be substituted in place of the shared disks discussed above; a controller corresponding to the disk controllers 470 and 480 would be used, and equipped with software modules to carry out the fencing operation.

In addition, the nodes, i.e. processor-based systems, that are members of the cluster can be any of a variety of processor-based devices, and in particular need not specifically be personal computers or workstations, but may be other processor-driven devices capable of issuing access requests to peripheral devices such as shared disks.

What is claimed is:

1. A method for preventing access to a shared peripheral device by a processor-based node in a multinode system, including the steps of:

(1) storing at the peripheral device a first unique value representing a first configuration of the multinode system;

(2) sending an access request from the node to the device, the request including a second unique value representing a second configuration of the multi-node system;

(3) determining whether said first and second values are identical; and

(4) if the first and second values are identical, then executing the access request at the peripheral device.

2. The method of claim 1, wherein:

said first value is generated utilizing at least in part information relating to a first time when the multinode system was in said first configuration; and

said second value is generated utilizing at least in part information relating to a second time when the multinode system was in said second configuration.

3. The method of claim 2, wherein:

step 3 includes the step of determining whether said first and second times are identical.

4. The method of claim 1, wherein said first and second values are generated based at least in part on epoch numbers generated by a membership protocol executing on said multinode system.

5. The method of claim 4, wherein each of said first and second values is generated based at least in part on respective membership sets of said multinode system generated by said membership protocol.

6. The method of claim 1, wherein each of said first and second values is generated based at least in part on respective membership sets of said multinode system generated by said membership protocol.

7. An apparatus for preventing access to at least one shared peripheral resource by a processor-based node in a multinode system, the resource being coupled to the system by a resource controller including a controller memory, each of a plurality of nodes on the system including a processor coupled to a node memory storing program modules configured to executing functions of the invention, the apparatus including:

a membership monitor module configured to determine a membership list of the nodes, including said resource, on the system at predetermined times, including at least at a time when the membership of the system changes;

a resource manager module configured to determine when the resource is in a failed state and for communicating the failure of the resource to said membership monitor to indicate to the membership monitor to generate a new membership list;

a configuration value module configured to generate a unique value based upon said new membership list and to store said unique value locally at each node on the system; and

an access control module stored at said controller memory configured to block access requests by at least one said requesting node to said resource when the locally stored unique value at said requesting node does not equal the unique value stored at said resource controller.

8. The apparatus of claim 7, wherein said configuration value monitor module is configured to determine said unique value based at least in part upon a time stamp indicating the time at which the corresponding membership list was generated.

9. The apparatus of claim 7, wherein said unique value is based at least in part upon an epoch number generated by a membership protocol module.

10. The apparatus of claim 7, wherein said membership monitor module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

11. The apparatus of claim 7, wherein said resource manager module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

12. The apparatus of claim 7, wherein said configuration module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

13. The apparatus of claim 7, wherein said access control module is configured to execute independently of any action by said shared resource when said shared resource is in a failed state.

Abstract of the Disclosure

A method and apparatus for fast and reliable fencing of resources such as shared disks on a networked system. For each new configuration of nodes and resources on the system, a membership program module generates a new membership list and, based upon that, a new epoch number uniquely identifying the membership correlated with the time that it exists. A control key based upon the epoch number is generated, and is stored at each resource controller and node on the system. If a node is identified as failed, it is removed from the membership list, and a new epoch number and control key are generated. When a node sends an access request to a resource, the resource controller compares its locally stored control key with the control key stored at the node (which is transmitted with the access request). The access request is executed only if the two keys match. The membership list is revised based upon a node's determination (by some predetermined criterion or criteria, such as slow response time) of the failure of a resource, and is carried out independently of any action (either hardware or software) of the failed resource.

Figure 1

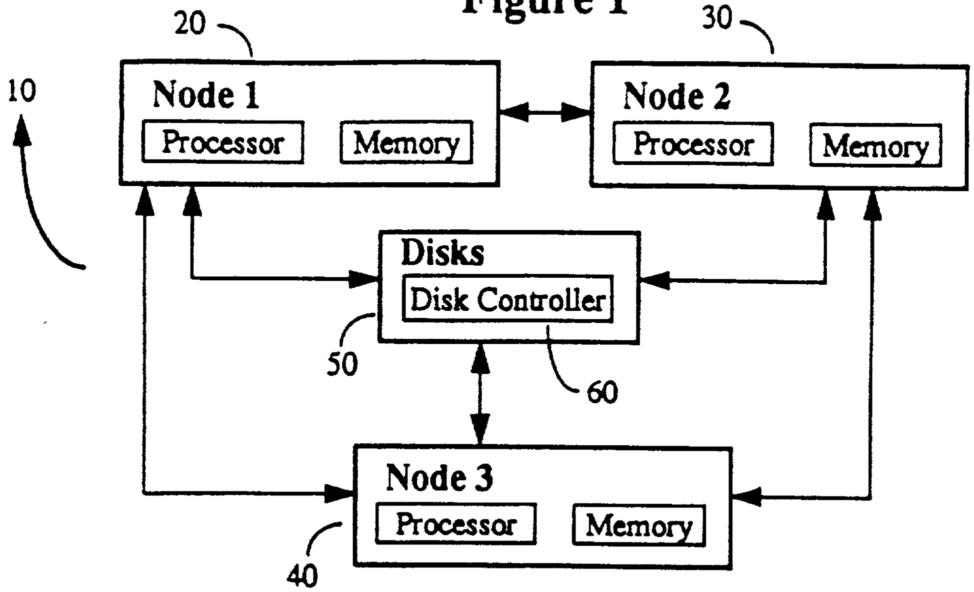
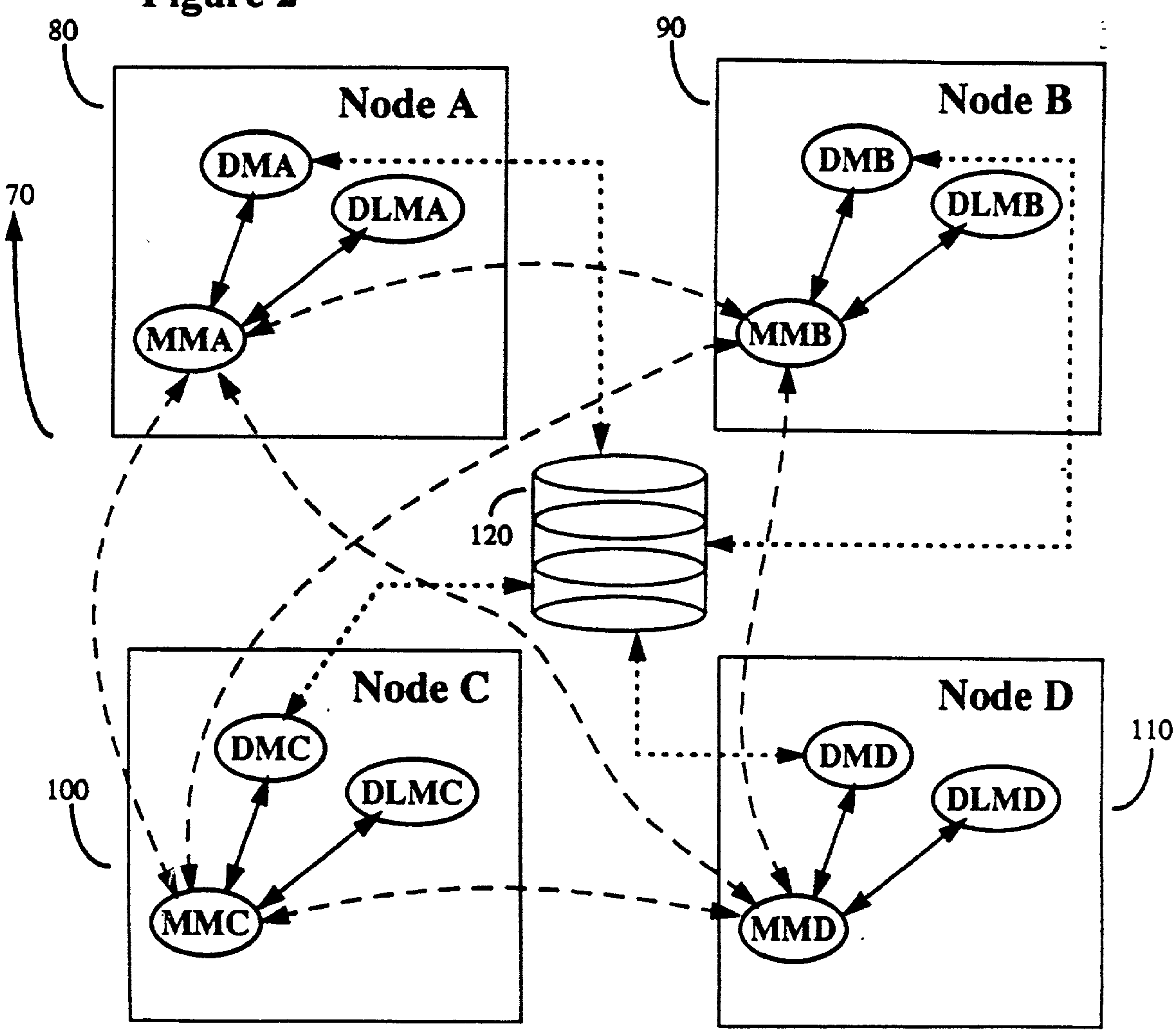


Figure 2



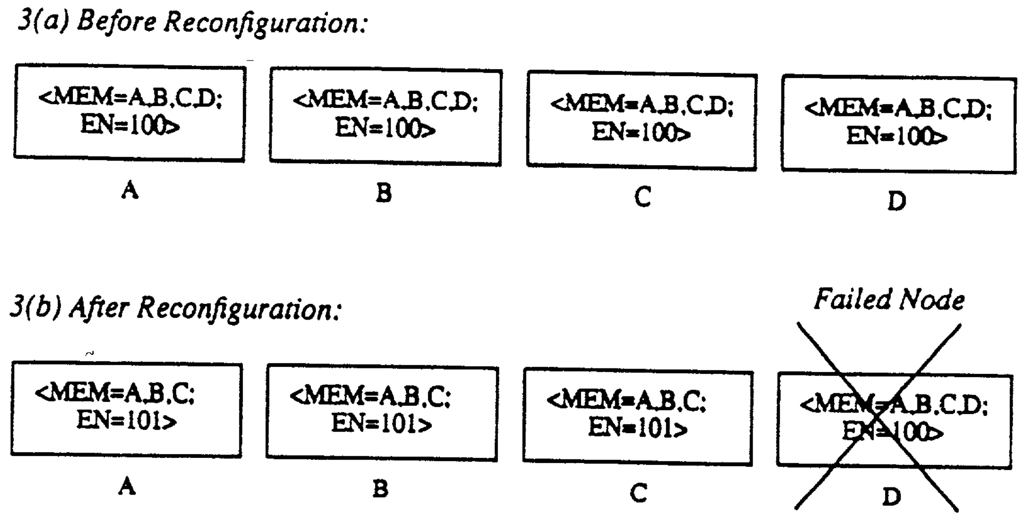


Figure 3

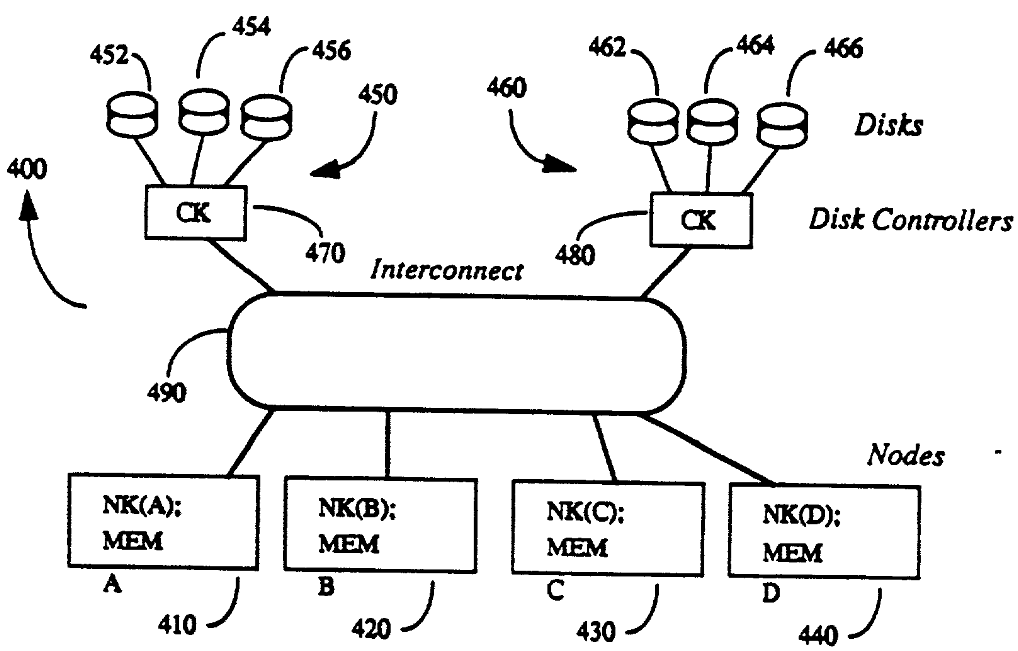


Figure 4

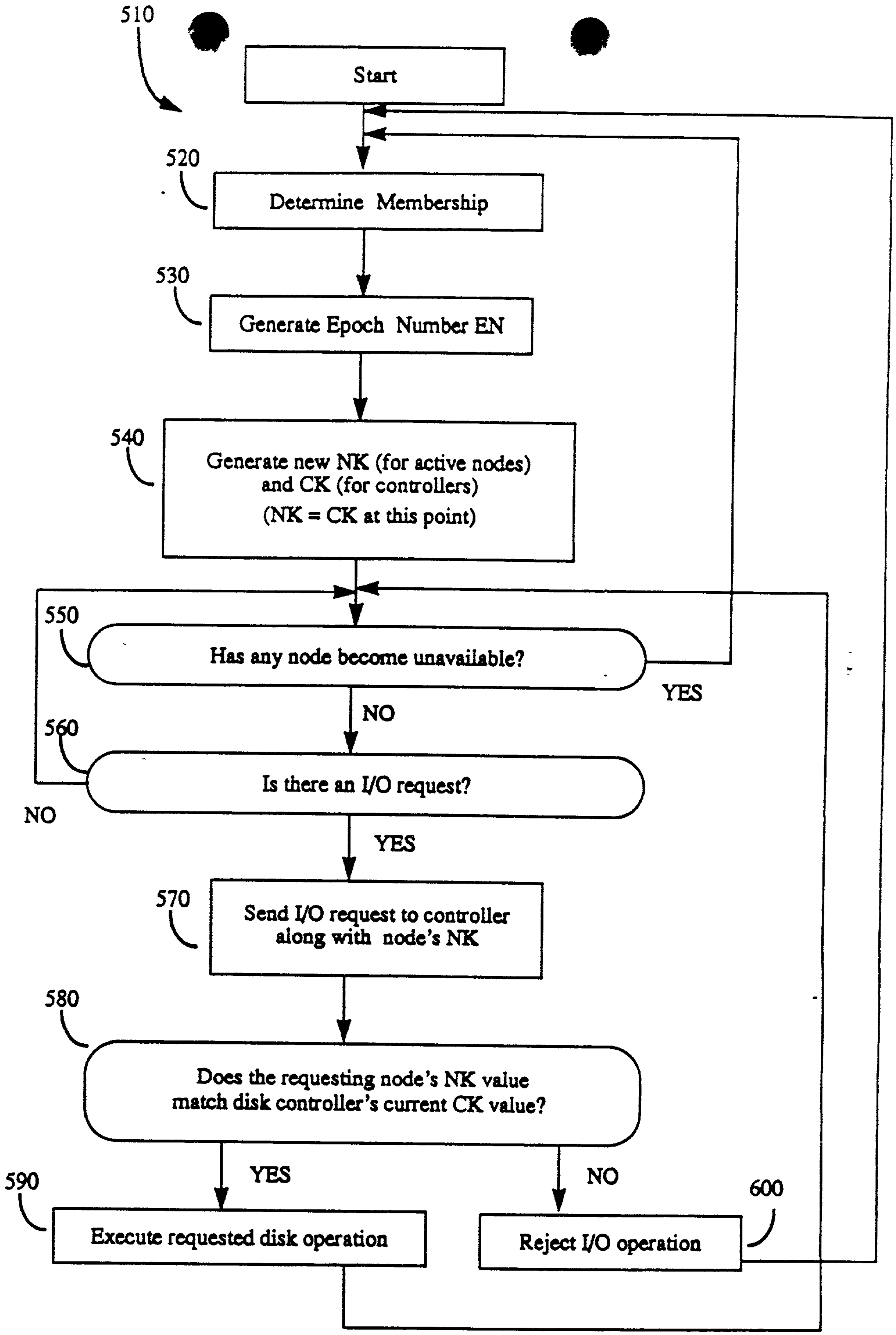

510
Start
520
Determine Membership
530
Generate Epoch Number EN
540
Generate new NK (for active nodes)
and CK (for controllers)
(NK = CK at this point)
550
Has any node become unavailable?
YES
NO
560
Is there an I/O request?
NO
YES
570
Send I/O request to controller
along with node's NK
580
Does the requesting node's NK value
match disk controller's current CK value?
YES
NO
590
Execute requested disk operation
600
Reject I/O operation

Figure 5

Attorney Docket No. 82225.P758

DECLARATION AND POWER OF ATTORNEY FOR SUN MICROSYSTEMS, INC.

UTILITY PATENT APPLICATION

OIPE JC68
NOV 2 0 2000
PATENT & TRADEMARK OFFICE

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND APPARATUS FOR RELIABLE DISK FENCING IN A MULTICOMPUTER SYSTEM

RECEIVED
NOV 2 4 2000
Technology Center 2100

the specification of which

X is attached hereto.
__ was filed on ____________________ as
__ Application Serial No. ____________________
and was amended on ____________________
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I do not know and do not believe that the same was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months prior to this application.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119, of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

rev. 8/10/94

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)	(Filing Date)	(Status - patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status - patented, pending, abandoned)

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

Keith G. Askoff, Reg. No. 33,828; Aloysius T.C. AuYeung, Reg. No. 35,432; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernardicou, Reg. No. P-35,934; Roger W. Blakely, Jr., Reg. No. 25,831; William D. Davis, Reg. No. 38,428; Daniel M. De Vos, Reg. No. 37,813; Gary B. Goates, Reg. No. 35,159; Scot A. Griffin, Reg. No. 38,167; David R. Halvorson, Reg. No. 33,395; Brian D. Hickman, Reg. No. 35,894; George W. Hoover, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; Dag H. Johansen, Reg. No. 36,172; Stephen L. King, Reg. No. 19,180;Daniel C. Mallery; Michael J. Mallie, Reg. No. 36,591; Kimberly G. Nobles, Reg. No. 38,255; Ronald W. Reagin, Reg. No. 20,340; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. P-39,018; James C. Scheller, Reg. No. 31,195; Edward W. Scott, IV, Reg. No. 36,000; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Allan T. Sponseller, Reg. No. 38,318; John C. Stattler, Reg. No. 36,285; David R. Stevens, Reg. 38,626; Edwin H. Taylor, Reg. No. 25,129; Lester J. Vincent, Reg. No. 31,460; Ben J. Yorks, Reg. No. 33,609; Norman Zafman, Reg. No. 26,250; Thomas X. Li, Reg. No. 37,079; and Edwin A. Sloane, Reg. No. 34,728 of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN and Lee Patch, Reg. No. 30,095; Erwin J. Basinski, Reg. No. 34,773; James W. Rose, Reg. No. 34,239; Kang S. Lim, Reg. No. 37,491; Matthew C. Rainey, Reg. No. 32,291; Timothy J. Crean, Reg. No. 37,116, and Leland Z. Wiesner of SUN MICROSYSTEMS, INC.

Address all correspondence to:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard, 7th Floor
Los Angeles, California 90025

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Vladimir Matena

Inventor's Signature Vladimir Matena   Date 11/1/95

Residence 1322 Kentfield Avenue, Redwood City, CA 94061   Citizenship Czech Republic

Post Office Address 1322 Kentfield Avenue, Redwood City, CA 94061

12. ☐ A verified statement claiming small entity status

☐ is enclosed or ☐ is on file in the prior application.

13. **X** The power of attorney in the prior application is to at least one of the following: FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P., Douglas B. Henderson, Reg. No. 20,291; Ford F. Farabow, Jr., Reg. No. 20,630; Arthur S. Garrett, Reg. No. 20,338; Donald R. Dunner, Reg. No. 19,073; Brian G. Brunsvold, Reg. No. 22,593; Tipton D. Jennings, IV, Reg. No. 20,645; Jerry D. Voight, Reg. No. 23,020; Laurence R. Hefter, Reg. No. 20,827; Kenneth E. Payne, Reg. No. 23,098; Herbert H. Mintz, Reg. No. 26,691; C. Larry O'Rourke, Reg. No. 26,014; Albert J. Santorelli, Reg. No. 22,610; Michael C. Elmer, Reg. No. 25,857; Richard H. Smith, Reg. No. 20,609; Stephen L. Peterson, Reg. No. 26,325; John M. Romary, Reg. No. 26,331; Bruce C. Zotter, Reg. No. 27,680; Dennis P. O'Reilley, Reg. No. 27,932; Allen M. Sokal, Reg. No. 26,695; Robert D. Bajefsky, Reg. No. 25,387; Richard L. Stroup, Reg. No. 28,478; David W. Hill, Reg. No. 28,220; Thomas L. Irving, Reg. No. 28,619; Charles E. Lipsey, Reg. No. 28,165; Thomas W. Winland, Reg. No. 27,605; Basil J. Lewris, Reg. No. 28,818; Martin I. Fuchs, Reg. No. 28,508; E. Robert Yoches, Reg. No. 30,120; Barry W. Graham, Reg. No. 29,924; Susan Haberman Griffen, Reg. No. 30,907; Richard B. Racine, Reg. No. 30,415; Thomas H. Jenkins, Reg. No. 30,857; Robert E. Converse, Jr., Reg. No. 27,432; Clair X. Mullen, Jr., Reg. No. 20,348; Christopher P. Foley, Reg. No. 31,354; John C. Paul, Reg. No. 30,413; David M. Kelly, Reg. No. 30,953; Kenneth J. Meyers, Reg. No. 25,146; Carol P. Einaudi, Reg. No. 32,220; Walter Y. Boyd, Jr., Reg. No. 31,738; Steven M. Anzalone, Reg. No. 32,095; Jean B. Fordis, Reg. No. 32,984; Roger D. Taylor, Reg. 28,992; Barbara C. McCurdy, Reg. No. 32,120; James K. Hammond, Reg. No. 31,964; Richard V. Burgujian, Reg. No. 31,744; J. Michael Jakes, Reg. No. 32,824; Thomas W. Banks, Reg. No. 32,719; Christopher P. Isaac, Reg. No. 32,616; Bryan C. Diner, Reg. No. 32,409; M. Paul Barker, Reg. No. 32,013; Andrew Chanho Sonu, Reg. No. 33,457; David S. Forman, Reg. No. 33,694; Vincent P. Kovalick, Reg. No. 32,867; James W. Edmondson, Reg. No. 33,871; Michael R. McGurk, Reg. No. 32,045; Joann M. Neth, Reg. No. 36,363; Gerson S. Panitch, Reg. No. 33,751; Cheri M.

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

Taylor, Reg. No. 33,216; Charles E. Van Horn, Reg. No. 40,266; Linda A. Wadler, Reg. No. 33,218; Jeffrey A. Berkowitz, Reg. No. 36,743; Michael R. Kelly, Reg. No. 33, 921; James B. Monroe, Reg. No. 33,971; Doris Johnson Hines, Reg. No. 34,629; Allen R. Jensen, Reg. No. 28,224; Lori Ann Johnson, Reg. No. 34,498; and David A. Manspeizer, Reg. No. 37,540; and SUN MICROSYSTEM, INC., Kenneth Olsen, Reg. No. 26,493, Timothy J. Crean, Reg. No. 37,116, Joseph T. FitzGerald, Reg. No. 33,881, Alexander E. Silverman, Reg. No. 37,940, Anirma Rakshpal Gupta, Reg. No. 38,275, Sean P. Lewis, Reg. No. 42,798, Michael J. Schallop, Reg. No. 44,319, Bernice B. Chen, Reg. No. 42,403, Kenta Suzue, Reg. No. 45,145, Noreen A. Krall, Reg. No. 39,734, Richard J. Lutton, Jr., Reg. No. 39,756, Marc Foodman, Reg. No. 34,110, and Naren Chaganti, Reg. No. 44,602.

14. **X** The power appears in the original declaration of the prior application.

15. ☐ Since the power does not appear in the original declaration, a copy of the power in the prior application is enclosed.

16. **X** Please address all correspondence to FINNEGAN, HENDERSON, FARABOW, GARRETT and DUNNER, L.L.P., 1300 I Street, N.W., Washington, D.C. 20005-3315.

17. ☐ Recognize as associate attorney ____________________

____

(name, address & Reg. No.)

LAW OFFICES
NNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
ASHINGTON, DC 20005
202-408-4000

<u>PETITION FOR EXTENSION</u>. If any extension of time is necessary for the filing of this application, including any extension in the parent application, serial no. 08/552,316, filed November 2, 1995, for the purpose of maintaining copendency between the parent application and this application, and such extension has not otherwise been requested, such an extension is hereby requested, and the Commissioner is authorized to charge necessary fees for such an extension to our Deposit Account No. 06-0916. A duplicate copy of this paper is enclosed for use in charging the deposit account.

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

By: Jeffrey H. Berkowitz
Jeffrey Berkowitz
Reg. No. 36,743

Date: November 20, 2000

09/721,222

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

PATENT
Attorney Docket No. 6502.0018

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
Vladimir MATENA	)	
Serial No.: 08/552,316	)	Group Art Unit: 2413
Filed: November 2, 1995	)	Examiner: Glenn Snyder
For: METHOD AND APPARATUS FOR RELIABLE DISK FENCING IN A MULTICOMPUTER SYSTEM	)	

OIPE JC68 NOV 2 0 2000 PATENT & TRADEMARK OFFICE

RECEIVED
NOV 2 4 2000
Technology Center 2100

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

REVOCATION OF ORIGINAL POWER OF ATTORNEY
AND GRANT OF NEW POWER OF ATTORNEY

Assignee Sun Microsystems, Inc., hereby revokes the previous Power of Attorney in the above action to Blakely Sokoloff Taylor & Zafman and hereby grant their power of attorney to Kenneth Olsen, Reg. No. 26,493; Matthew C. Rainey, Reg. No. 32,291; Erwin J. Basinski, Reg. No. 34,773; Timothy J. Crean, Reg. No. 37,116; Leland Z. Wiesner, Reg. No. 39,424; Philip J. McKay, Reg. No. 38,966; and to **FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P., Reg. No. 22,540,** Douglas B. Henderson, Reg. No. 20,291; Ford F. Farabow, Jr., Reg. No. 20,630; Arthur S. Garrett, Reg. No. 20,338; Donald R. Dunner, Reg. No. 19,073; Brian G. Brunsvold, Reg. No. 22,593; Tipton D. Jennings IV, Reg. No. 20,645; Jerry D. Voight, Reg. No. 23,020; Laurence R. Hefter, Reg. No. 20,827; Kenneth E. Payne, Reg. No. 23,098; Herbert H. Mintz, Reg. No. 26,691; C. Larry

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

O'Rourke, Reg. No. 26,014; Albert J. Santorelli, Reg. No. 22,610; Michael C. Elmer, Reg. No. 25,857; Richard H. Smith, Reg. No. 20,609; Stephen L. Peterson, Reg. No. 26,325; John M. Romary, Reg. No. 26,331; Bruce C. Zotter, Reg. No. 27,680; Dennis P. O'Reilley, Reg. No. 27,932; Allen M. Sokal, Reg. No. 26,695; Robert D. Bajefsky, Reg. No. 25,387; Richard L. Stroup, Reg. No. 28,478; David W. Hill, Reg. No. 28,220; Thomas L. Irving, Reg. No. 28,619; Charles E. Lipsey, Reg. No. 28,165; Thomas W. Winland, Reg. No. 27,605; Basil J. Lewris, Reg. No. 28,818; Martin I. Fuchs, Reg. No. 28,508; E. Robert Yoches, Reg. No. 30,120; Barry W. Graham, Reg. No. 29,924; Susan Haberman Griffen, Reg. No. 30,907; Richard B. Racine, Reg. No. 30,415; Thomas H. Jenkins, Reg. No. 30,857; Robert E. Converse, Jr., Reg. No. 27,432; Clair X. Mullen, Jr., Reg. No. 20,348; Christopher P. Foley, Reg. No. 31,354; John C. Paul, Reg. No. 30,413; Roger D. Taylor, Reg. No. 28,992; David M. Kelly, Reg. No. 30,953; Kenneth J. Meyers, Reg. No. 25,146; Carol P. Einaudi, Reg. No. 32,220; Walter Y. Boyd, Jr., Reg. No. 31,738; Steven M. Anzalone, Reg. No. 32,095; Jean B. Fordis, Reg. No. 32,984; Barbara C. McCurdy, Reg. No. 32,120; James K. Hammond, Reg. No. 31,964; Richard V. Burgujian, Reg. No. 31,744; J. Michael Jakes, Reg. No. 32,824; and Jennifer M. Orzech, Reg. No. 40,112; both jointly and separately as their attorneys with full power of substitution and revocation to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith, and to receive the Letters Patent.

LAW OFFICES
INNEGAN, HENDERSON,
FARABOW, GARRETT
& DUNNER, L.L.P.
1300 I STREET, N. W.
VASHINGTON, DC 20005
202-408-4000

Please send all future correspondence concerning this application to Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P. at the following address:

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005-3315

Date: 2/6/97

Kenneth Olsen
Vice President Intellectual Property
Sun Microsystems, Inc.